WHAT IS CLAIMED IS:

Sub ~~B5~~ D1

1. A method of controlling the execution of instructions within a pipelined processor, comprising:

providing an instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said words comprising a jump instruction;

assigning one of a plurality of values to at least one of said data bits of said at least one jump instruction; and

controlling the execution of at least one subsequent instruction within said pipeline based on said one assigned value of said at least one data bit when said at least one jump instruction is decoded.

2. The method of Claim 1, wherein the act of assigning comprises: identifying a plurality of data bits within said at least one jump instruction; and assigning one of two discrete values to each of said data bits, the combination of said two discrete values representing at least three jump delay slot modes within said processor.

3. The method of Claim 2, wherein the act of controlling the execution based on said discrete values comprises selecting at least one mode from the group comprising:

(i) executing said at least one subsequent instruction under all circumstances;

(ii) executing said at least one subsequent instruction only if a jump occurs;

and (iii) stalling the pipeline or inserting a bubble into the pipeline if a jump occurs.

4. The method of Claim 3, wherein said at least one jump instruction comprises a conditional branch instruction.

5. The method of Claim 1, wherein the act of controlling the execution based on said one assigned value comprises:

Cont D1

(i) executing said at least one subsequent instruction under all circumstances;

(ii) executing said at least one subsequent instruction only if a jump occurs; and

(iii) stalling the pipeline or inserting a bubble into the pipeline if a jump occurs.

6. A processor design synthesized by the method comprising:

inputting information to a first file to include an instruction set having at least one jump instruction, said at least one jump instruction comprising at least one of a plurality of jump delay modes;

defining the location of at least one library file;

generating a script using said first file, said library file, and user input information;

running said script to create a customized description language model; and

synthesizing said design based on said description language model.

7. The method of Claim 6, wherein the act of synthesizing comprises running synthesis scripts based on said description language model.

8. The method of Claim 7, further comprising the act of generating a third file for use with a simulation, and simulating said design using said third file.

9. The method of Claim 8, further comprising the act of evaluating the acceptability of the design based on said simulation.

10. The method of Claim 9, further comprising the acts of revising the design to produce a revised design, and re-synthesizing said revised design.

11. The method of Claim 1, wherein the act of inputting comprises providing a plurality of input parameters associated with said design, said parameters comprising:

(i) a cache configuration; and

(ii) a memory interface configuration.

12. A machine readable data storage device comprising:

a data storage medium adapted to store a plurality of data bits; and

a computer program rendered as a plurality of data bits and stored on said data storage medium, said program being adapted to run on the processor of a computer system and synthesize integrated circuit logic for use in a processor having a pipeline and incorporating an instruction set having a at least one branching instruction and a plurality of jump modes associated therewith, said plurality of jump modes comprising at least the following:

(i) executing a subsequent instruction within said pipeline under all circumstances;

(ii) executing a subsequent instruction within said pipeline only if jumping occurs; and

(iii) stalling said pipeline if jumping occurs.

13. The data storage device of Claim 12, wherein said data storage medium is a compact disk-read only memory (CD-ROM), and said plurality of data bits comprises an object representation of said program.

Sub B6 D1

14. A digital processor comprising:

a processor core having a multistage instruction pipeline, said core being adapted to decode and execute an instruction set comprising a plurality of instruction words;

a data interface between said processor core and an information storage device; and

an instruction set comprising a plurality of instruction words, at least one of said instruction words being a jump instruction containing data defining a plurality of jump delay slot modes, said plurality of modes controlling the execution of instructions within said instruction pipeline of said processor core in response to said at least one jump instruction word within said instruction set.

15. The processor of Claim 14, wherein said plurality of jump delay slot modes comprises at least the following modes:

Sub D1

(i) executing a subsequent instruction within said pipeline under all circumstances;

(ii) executing a subsequent instruction within said pipeline only if jumping occurs; and

(iii) stalling the pipeline if jumping occurs.

16. The processor of Claim 14, wherein said at least one jump instruction comprises a conditional branch instruction having an associated logical condition, the execution of a jump to the address within said information storage device specified by said at least one conditional branch instruction being determined by said logical condition.

~~Sub B2~~

17. A digital processor having at least one pipeline and an associated data storage device, wherein the execution of instructions within said at least one pipeline is controlled by the method comprising:

storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a branch instruction directing branching to a first address within said data storage device;

assigning one of a plurality of values to at least one of said data bits of said at least one branch instruction;

decoding said at least one branch instruction including said one value;

determining whether to execute an instruction within said pipeline in a stage preceding that of said at least one branch instruction based on said one value; and

branching to said first address based on said at least one branching instruction.

18. The processor of Claim 17, wherein said data bits comprise binary (base 2) data.

19. The method of Claim 17, wherein said at least one pipeline comprises at least a three stage instruction pipeline comprising instruction fetch, decode, and execute stages.

01 Sub B5

20. A method of controlling the branching within the program of a multi-stage pipelined digital processor, comprising:

storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a branch instruction directing branching to a first address within said data storage device based on a first parameter;

defining a plurality of jump modes;

assigning at least one of said plurality of jump modes to at least one of said data bits of said at least one branch instruction;

decoding said at least one branch instruction including said at least one data bit; and

determining whether to branch to said first address based on said at least one data bit and said first parameter.

21. The method of Claim 20, wherein the act of defining a plurality of jump modes comprises defining the following modes:

(i) executing a subsequent instruction under all circumstances;

(ii) executing a subsequent instruction only if jumping occurs; and

(iii) stalling the pipeline or inserting a bubble into the pipeline if jumping occurs.

22. An apparatus for synthesizing the design of logic used within a digital processor, comprising:

a central processing unit;

a data storage device operatively coupled to said central processing unit, said data storage device being adapted to store and retrieve a computer program;

an input device adapted to generate signals in response to inputs from a user of said system;

a computer program, stored on said data storage device, said program being adapted to receive said signals and permit said user to:

input information to a first file to include an instruction set having at least one jump instruction, said at least one jump instruction comprising at least one of a plurality of jump delay modes;

define the location of at least one library file;

generate a script using said first file, said library file, and user input information;

run said script to create a customized description language model; and

synthesize said design based on said description language model.

Sub B9

23. A digital processor comprising:

processing means having a multistage data pipeline, said processing means being adapted to decode and execute an instruction set comprising a plurality of instruction words;

means for storing data;

data interface means for transferring data between said processing means and said means for storing data; and

an instruction set comprising a plurality of instruction words, at least one of said instruction words being a jump instruction containing data defining a plurality of jump control means, said plurality of jump control means controlling the execution of instructions within said data pipeline of said processing means in response to said at least one jump instruction word within said instruction set.

24. An apparatus for synthesizing the design of logic used within a digital processor, comprising:

means for processing data;

means for storing data, said means for storing being operatively coupled to said means for processing, and adapted to store and retrieve a computer program;

means for inputting information operatively coupled to said means for processing, said means for inputting being adapted to generate signals in response to inputs from a user of said system;

a computer program, stored on said data storage device, said program being adapted to receive said signals and comprising:

means for inputting information to a first file to include an instruction set having at least one jump instruction, said at least one jump instruction comprising at least one of a plurality of jump delay modes;

means for defining the location of at least one library file;

means for generating a script using said first file, said library file, and user input information;

means for running said script to create a customized description language model; and

means for synthesizing said design based on said description language model.

Add A6

Add B10

Add E1